Status of the Claims:

Although the claims have not been amended, below is a listing of the claims as they now stand.

1. (Original) A method of detecting a computer virus that attempts to gain access to restricted computer system resources, comprising:

emulating computer executable code in a subject file; and

monitoring the emulation of the computer executable code and monitoring a memory state of the computer system for modifications caused by the emulated instructions in the computer executable code, to detect an attempt by the emulated code to access one or more of the restricted computer system resources.

2. (Original) The method of claim 1, wherein monitoring the emulation includes detecting installation of new exception handler followed by forcing of a corresponding exception.

3. (Original) The method of claim 1, wherein monitoring the emulation includes detecting writing of a new pointer to at least one predetermined address in system memory for storing an exception handler pointer.

4. (Original) The method of claim 1, wherein monitoring the emulation includes detecting installation, in system memory, of a new pointer to an exception handler.

5. (Original) The method of claim 1, wherein monitoring the emulation includes detecting installation of a new interrupt handler followed by forcing of a corresponding interrupt.

6. (Original) The method of claim 1, wherein monitoring the emulation includes detecting writing of a new pointer to at least one predetermined address in system memory for storing an interrupt handler pointer.

7. (Original) The method of claim 1, wherein monitoring the emulation includes detecting use of a predetermined instruction to retrieve an address in system memory corresponding to an interrupt descriptor table.

8. (Original) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for detecting a computer virus that attempts to gain access to restricted computer system resources, the method steps comprising:

emulating computer executable code in a subject file; and

monitoring the emulation of the computer executable code and monitoring a memory state of the computer system for modifications caused by the emulated instructions in the computer executable code, to detect an attempt by the emulated code to access one or more of the restricted computer system resources.

9. (Original) A computer system, comprising:

a processor; and

a program storage device readable by the computer system, tangibly embodying a program of instructions executable by the processor to perform method steps for detecting a computer virus that attempts to gain access to restricted computer system resources, the method steps comprising:

emulating computer executable code in a subject file; and

monitoring the emulation of the computer executable code and monitoring a memory state of the computer system for modifications caused by the emulated instructions in the computer executable code, to detect an attempt by the emulated code to access one or more of the restricted computer system resources.

10. (Original) A computer data signal embodied in a transmission medium which embodies a program of instructions executable by a computer for detecting a computer virus that attempts to gain access to restricted computer system resources, comprising:

a first segment including emulation code to emulate computer executable code in a subject file; and

a second segment including monitor code to monitor emulation of the computer executable code and monitoring a memory state of the computer system for modifications caused by the emulated instructions in the computer executable code; and

a third segment including detector code to detect an attempt by the emulated code to access one or more of the restricted computer system resources.

11. (Original) An apparatus for detecting computer viruses that attempt to gain access to restricted computer system resources, comprising:

an emulator component, wherein the emulator component emulates computer executable code in a subject file;

a monitor component, wherein the monitor emulation of the computer executable code and monitoring a memory state of the computer system for modifications caused by the emulated instructions in the computer executable code, and supplies information regarding the emulated code and modification of the memory state; and

a detector component, wherein the detector component, based on the information supplied by the monitor component regarding the emulated code execution and modification of memory state by the emulated code execution, detects an attempt by the emulated code to access one or more of the restricted computer system resources.

12. (Original) The apparatus of claim 11, wherein the monitor component monitors system memory.

13. (Original) The apparatus of claim 11, wherein the detector component detects installation of a new exception handler.

14. (Original) The apparatus of claim 13, wherein after the detector component detects installation of a new exception handler, the detector component monitors code execution to detect forcing of a corresponding exception.

15. (Original) The apparatus of claim 11, wherein the detector component detects writing of a new pointer to at least one predetermined address in system memory for storing an exception handler pointer.

16. (Original) The apparatus of claim 11, wherein the detector component detects installation of a new interrupt handler.

17. (Original) The apparatus of claim 16, wherein after the detector component detects installation of a new interrupt handler, the detector component monitors code execution to detect forcing of a corresponding interrupt.

18. (Original) The apparatus of claim 11, wherein the detector component detects writing of a new pointer to at least one predetermined address in system memory for storing an interrupt handler pointer.

19. (Original) The apparatus of claim 11, wherein the monitor component detects use of a predetermined instruction to retrieve an address in system memory corresponding to an interrupt descriptor table.